Comprehensive Investigations on QUEST: a Novel QoS-Enhanced Stochastic Packet Scheduler for Intelligent LTE Routers

+Suman Paul[1,2], #Malay Kumar Pandit[1]
[1] Dept. of ECE, Haldia Institute of Technology, Haldia,
West Bengal 721657 - INDIA
[e-mail: +paulshit61@gmail.com, #mkpandit.seci@gmail.com]
[2] School of Engineering and Technology, West Bengal University of Technology (Maulana Abul Kalam Azad University of Technology West Bengal)
West Bengal 700064 - INDIA
*Corresponding author : Suman Paul

*Received December 7, 201; revised April 19, 2017; revised June 13, 2017; revised July 24, 2017;
revised August 27, 2017; accepted June 22, 2017; published February 28, 2018*

Abstract

In this paper we propose a QoS-enhanced intelligent stochastic optimal fair real-time packet scheduler, QUEST, for 4G LTE traffic in routers. The objective of this research is to maximize the system QoS subject to the constraint that the processor utilization is kept nearly at 100 percent. The QUEST has following unique advantages. First, it solves the challenging problem of starvation for low priority process - buffered streaming video and TCP based; second, it solves the major bottleneck of the scheduler Earliest Deadline First’s failure at heavy loads. Finally, QUEST offers the benefit of arbitrarily pre-programming the process utilization ratio.Three classes of multimedia 4G LTE QCI traffic, conversational voice, live streaming video, buffered streaming video and TCP based applications have been considered. We analyse two most important QoS metrics, packet loss rate (PLR) and mean waiting time. All claims are supported by discrete event and Monte Carlo simulations. The simulation results show that the QUEST scheduler outperforms current state-of-the-art benchmark schedulers. The proposed scheduler offers 37 percent improvement in PLR and 23 percent improvement in mean waiting time over the best competing current scheduler Accuracy-aware EDF.

Keywords: Cache and deadline misses, hidden Markov model (HMM), Long Term Evolution (LTE), packet loss rate (PLR), quality of service (QoS), scheduler.

http://doi.org/10.3837/tiis.2018.02.003 **ISSN : 1976-7277**

1. Introduction

QoS (Quality of service) is defined [1] as the collective effect of service performance which determines the degree of satisfaction of a user of the service. QoS requirements have always posed a challenge from scheduling perspective. In telecommunication systems QoS is directly related to the network performance of the underlying routing systems. The primary goal of QoS is to provide priority including dedicated bandwidth, guranteed throughput, controlled jitter and latency (real-time interactive traffic) and to minimize the packet loss. In search for quality, current researchers are trying to maximize QoS of real-time embedded systems including routers. A router is a specific case of soft-real time embedded systems. Scheduling of different classes of traffic (both GBR and non-GBR) is a *crucial* integral part of modern LTE routers. Optimally scheduling the different tasks in a multitasking computing system is important. Optimizing the system performance and achieving high processor utilization depends on appropriate processor usage time allocated to the processes for guaranteeing high system class-specific QoS. The system QoS is of prime concern in designing state-of-the-art real-time embedded systems e.g., routers as it addresses key network parameters like throughput, sources of errors, resource availabilities, fair bandwidth allocation, latencies (sum of mean waiting time and service time), packet loss rate (PLR), end-to-end delay, jitter (delay variation), etc. In this work we propose and investigate on a *probabilistic framework* for a novel *optimal intelligent real-time* embedded computing scheduler, QUEST (quality-of-service enhanced stochastic), for *routers of 4G Long Term Evolution (LTE)* [2] traffic classes. We address following gaps in scheduler research. One is the starvation of low priority class - buffered streaming video & TCP based application processes. Further, we address the poor performance of the premier Earliest Deadline First (EDF) scheduler at heavy network traffic loads. Addressing these two problems motivated us to undertake the present research. In EDF scheduler and its variants, the rise of the mean waiting time to an unacceptably high level at heavy traffic loads, is a crucial problem which has been solved in this work by concentrating on the heavy-load zone when the processor utilization is close to 100 percent.

The objective of our research is to design of a QoS-enhanced intelligent stochastic real-time packet scheduler, QUEST for LTE class [3] traffic which satisfies the QoS requirements defined by the QoS architecture of 3GPP specifications [4].

A. *Scheduling Attributes*

The proposed pre-emptive real-time schedulier differs from the conventional pre-emptive schedulers in that it is probabilistic in nature in order to keep the utilization fixed at nearly 100 pecent in a fair way and offers the following *novelties*:

(i) Current researchers have just proposed various methods for maximizing processor utilization in network routers which are QoS-aware and have enhanced QoS only *qualitatively*. But in this research for the first time, fixing utilization close to 100 percent, QoS is *quantitatively* maximized using machine learning.

(ii) Lower priority class processes (buffered streaming video & TCP based applications) attain a guaranteed minimum amount of processor time due to the pre-designed distribution of individual process utilization. The scheduler is fair to all of its traffic and eliminates the problem of priority starvation.

(iii) The adaptability and re-configurability of the QUEST scheduler has been implemented using a *machine-learning* feedback controller. The feedback-controller with the help of run-time cache-miss and deadline-miss error feedbacks learns and takes corrective decisions to maximize the system QoS.

(iv) QUEST is strongly immune from hacking because the scheduler is random in nature and therefore the next process to be executed cannot be predicted apriori.

(v) The objective is to maximize the system QoS, subject to the constraint that utilization is kept nearly at 100 percent. An optimum utilization of 100 percent is *enforced*. In this scheduling scheme, process utilization, U_i for a process P_i, is expressed as,

$$U_i = \frac{T_i}{D_i} \tag{1}$$

where T_i is the fraction of time spent for execution of process $P_{i.}$ D_i is denoted as the deadline of the process $P_{i.}$ The state probability vector of process utilization ratio of n processes running in a system can be expressed as, $\prod$

$$\prod=[U_1:U_2:..U_{n-1}:U_n] \tag{2}$$

The proposed scheduler is dynamic priority based. In Section 6.3, it is demonstrated that $\sum U_i=1$, which indicates that the processor utilization is 100 percent. 100 percent utilization ensures that the scheduler is *optimally* schedulable [5].

In practice, for an end-to-end QoS sensitive LTE traffic, which has a commitment to deliver on time, the process utilization for different service classes traffic is tailored in such a way that a guaranteed minimum amount of processor attention (time) for each class traffic is maintained. For multimedia embedded (4G LTE router) applications considered in this paper, Conversational voice (GBR), Interactive live streaming video (GBR) and buffered streaming video & TCP based applications (non-GBR) processes follow a long-tailed *Pareto* distribution of process utilization ratio. By running QUEST, a *target* process utilization ratio is maintained as per designer’s requirement. In this work, a practical case of process utilization ratio, U_i, for three processes has been provisioned in the ratio of 80:16:4.

B. System Quality of Service (QoS)

Delivering QoS refers to guaranteeing given service parameters within certain bounds for a network. The most dominant QoS parameter, packet loss rate (PLR) in a router is encountered in system activities that may arise due to different errors like deadline miss, L1 and L2 cache misses, page fault, etc. In this paper we focus two most important QoS’s metrics, namely, PLR and mean waiting time (related to system latency). Practical cache miss error probabilities come in the range of [10^{-2}-10^{-1}] [6,7]. Practical deadline miss error probabilities come in the range of [0.013 - 0.12] [8]. For practical real-time tasks, the deadline for 4G LTE traffic varies in the range of 10-300 ms [9,10].

2. Related Work

Next generation networks are required to transport and manage a wide range of applications with diverse traffic class QoS requirements and features. 3GPP has developed a QoS framework and a set of radio resource management techniques. However, 3GPP technical specifications do not define any specific scheduling algorithms to support real time and non-real time service classes. As a result, in the last few years, a variety of distinguished classical scheduling algorithms have been proposed and investigated. These includes the basic Proportional Fair, Round Robin, Maximal Signal-to-Interference Ratio, and Fair Throughput and complex schemes like exponential rule, intra-class, and inter-class schedulers for LTE traffic [11-13]. These schemes combine one or two of the scheduling criteria but none of them considers the LTE service class attributes related to process utilization. The authors in [14-16], have proposed and identified a Multi-level scheduling schemes which involves distributing the task of scheduling into different stages. Marinčić and Šimunić in [17], have made a comparative performance analysis of different scheduling algorithms in LTE system: Round Robin, Proportional Fair, Best CQI, Resource Fair and MaxMin. The authors in [18], have proposed a LTE downlink sacheduling scheme for voice services based on user perception. The proposed scheduler is not a reconfigurable one. Further the scheduler is not utilization driven. In routers, the simplest First-come first-served (FCFS) scheduler receives packets from all input traffic classes. Packets are assigned to a single queue upon arrival and are serviced on a first-come, first-served basis. An FCFS scheduler cannot differentiate traffic classes. Packets may be dropped if the queue is full. In [19], Toral-Cruz *et al.* have analyzed QoS parameters, namely, jitter and packet loss rate of VoIP traffic. The studies have revealed that VoIP jitter can be modeled by self-similar processes with short or long range dependences. However, the reserch has no specific focus on maximizing the performance of QoS metrics. In [20], Cristofaro *et al*. have presented a comparative analysis of QoS attributes for the VoIP and videoconferencing traffic with different queueing policies. In [21], the authors have proposed a queuing delay control and adjustment method, which guarantees the required QoS in terms of per-service traffic flow authorized for the real-time multi-service traffic. This method deals how to control the queuing delay value at the specified waiting delay by adjusting the arrival probability, so that the QoS delay for real-time services may be guaranteed. Greco *et al.* [22] have worked on a multitasking, pre-emptive RTOS environment in a stochastic scheduling domain. The proposed scheduler is not a reconfigurable one. Although the model is based on Markov chain, the has no focus on state estimation by machine learning. The authors in [23], have proposed and evaluated various queueing disciplines, considering, priority queueing (PQ), fair queueing (FQ), custom queueing (CQ), low-latency queueing (LLQ) in IP routers to provision the end-to-end QoS requirements for various traffic flows. To meet the target QoS requirements, in case of increasing high priority traffic sources, the authors have proposed solution either by changing the prioritization scheme at the switching routers in favour of priority classes or by allocating more bandwidth. However, the scheme cannot eliminate the problem of priority starvation for low priority best effort traffic classes and allocation of bandwidth is not a dynamic one. Kooti *et al.* [24] have demonstrated a reconfiguration-aware real-time scheduling mechanism under QoS constraints where only VoIP traffic has been considered. Further, no explicit mechanism to enhance the system QoS and supporting queueing theory are not mentioned.

Based on literature survey it is observed that in a multitasking scheduler in LTE routers, dynamically optimizing the system QoS for Long Term Evolution (LTE) traffic based on Markov chain model has not been found in the literature. In this work, we have applied a novel search technique to find the global minimum value of PLR in the search space. Many authors have proposed various scheduling approaches based on real-time pre-emptive scheduling algorithms, for example static priority scheduling: rate monotonic (RM), dynamic priority scheduling: earliest deadline first (EDF) and its variants. In these scheduling mechanisms, lower priority processes suffer because of suspension of execution by the higher priority class processes (flows). Using EDF in a dynamic environment of real-world applications of an LTE enviornment for an overloaded system processes miss deadlines frequently resulting in very low value of throughput. Further, EDF is incompatible in real-time packet network traffic as all traffic classes receive the same miss rate irrespective of class-specific deadline requirements and traffic characteristics. Moreover, EDF does not meet the requirements of class differentiation for traffic and therefore fails to comply with the service level agreements (SLAs) with client processes. Last, EDF and its variant A-EDF are deadline driven, where process utilization has no explicit focus. The *root* of the problem can be traced to its deterministic and deadline-driven mode of operation. Taking a *novel* alternative *route* here, namely, non-deterministic stochastic and utilization (load)-driven operation, the problem has been solved.

The above mentioned problems have been solved using QUEST scheduler. In this framework, a non-deterministic optimal scheduler which is random in nature has been implemented for 4G LTE network traffic so that the highest priority class process, conversational voice does not dominate the processor execution time. As a result, the problem of starvation of the low priority flow of buffered streaming video and TCP based applications never occurs. QUEST is *traffic class-sensitive* and conforms to SLAs. Moreover, QUEST is a deadline-aware utilization-driven scheduler for 4G LTE traffic.

The rest of this paper is organized as follows. Sections 3 and 4 demonstrate proposed system model and formulate the scheduling mechanism and queue management, respectively. Section 5 states the simulation methodology and environment. Simulation results are reported in Section 6. Section 7 illustrates dynamic global optimization and re-configurability of the QUEST. Section 8 demonstrates the run-time estimation of transition probability matrix (TPM) by machine learning. A comparative performance analysis of the proposed scheduler is reported in Section 9. Experimental results of QUEST are reported in Section 10. Section 11 concludes the paper and summarizes the key findings.

3. Proposed System Model

The design for the scheduling framework has been implemented for three service classes: Conversational voice (GBR), Interactive live streaming video (GBR) and buffered streaming video & TCP based applications (non-GBR). A Finite-state machine (FSM) based on Markov model (Discrete-Time Markov process) for the scheduler is proposed and presented in this paper. A Discrete-Time Markov Process (DTMP) $\{X_n\}$ such its future state only depends on current state and it is independent of its state values in earlier time steps: n-1, n-2, etc. A DTMF defined over a discrete state space is known as Discrete-Time Markov Chain. Each process in this scheme modelled as a particular Markov state. The processes are characterized by their state probabilities (p_{ij})s which are defined as probabilities of processes to be in their own states

($p_{ij,i=j}$) or to make transitions to other states ($p_{ij,i\neq j}$). In this model, the processes settle to a steady state probability distribution according as time evolves.

The proposed model behind this scheduling framework is a Hidden Markov Model (HMM) [25]. Since HMM is an NP-Hard problem [25] Markov initial TPM parameters (matrix elements) are calculated *apriori* using machine learning Metropolis-Hastings algorithm [26] : stated in algorithm 1 represented by **Fig. 1**. Metropolis-Hastings algorithm is a special class of Markov Chain Monte Carlo (MCMC) method, with constraints like the diagonal elements of the TPM are in the range: [0.4 - 0.9] and the non-diagonal elements are in the range: [0.01 - 0.6] [27]. As a result a faster convergence is achieved in such cases. Because of Markovian property, target steady state probability distribution can be generated. The corresponding TPM is estimated by maximum likelihood. To support the above proposition in an embedded computing environment of an LTE router, the desired steady state probability distribution, for three class processes has been considered.

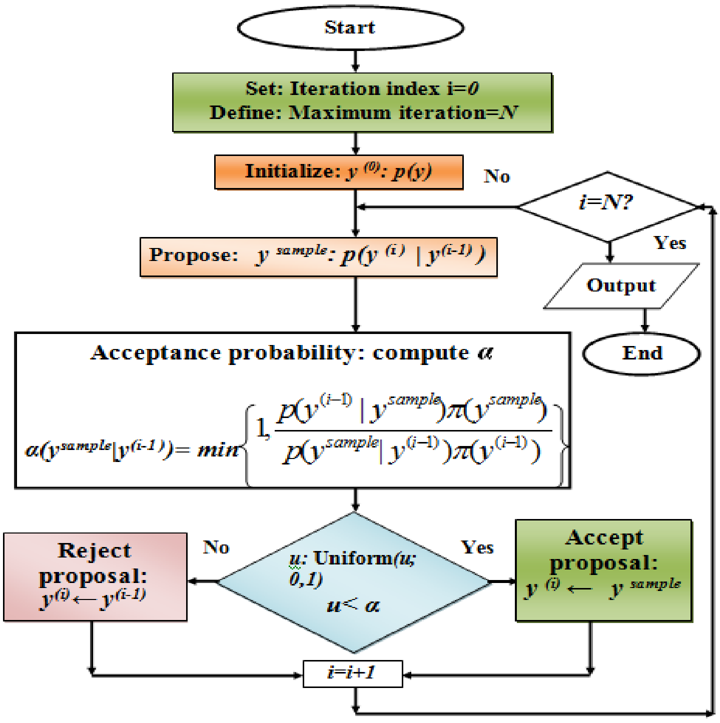
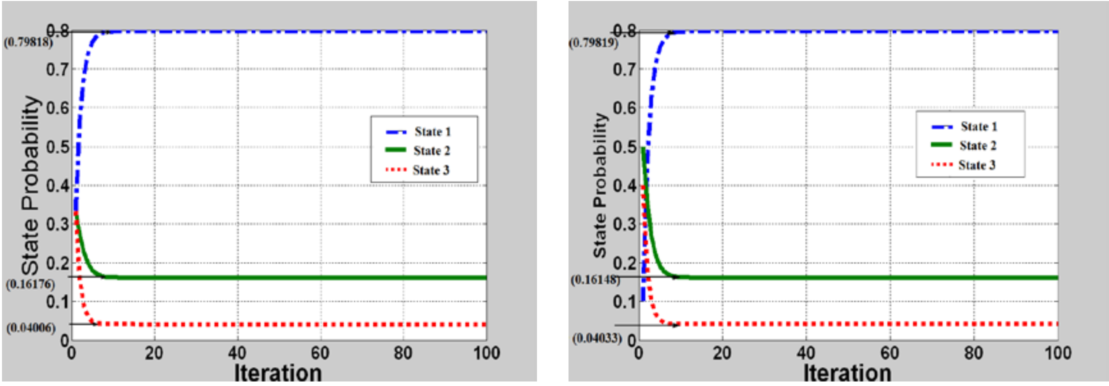

Start
Set: Iteration index i=0
Define: Maximum iteration=N
Initialize: $y^{(0)}: p(y)$
No
i=N?
Yes
Output
End
Propose: $y^{sample}: p(y^{(i)} \mid y^{(i-1)})$
Acceptance probability: compute α
$\alpha(y^{sample} \mid y^{(i-1)}) = min\left\{1, \frac{p(y^{(i-1)} \mid y^{sample})\pi(y^{sample})}{p(y^{sample} \mid y^{(i-1)})\pi(y^{(i-1)})}\right\}$
Reject proposal: $y^{(i)} \leftarrow y^{(i-1)}$
No
u: Uniform(u; 0,1)
u< α
Yes
Accept proposal: $y^{(i)} \leftarrow y^{sample}$
i=i+1

Fig. 1. Algorithm 1: Metropolis-Hastings algorithm

The first step is to initialize the sample value for each random variable. The algorithm consists of three steps: First, a proposal sample y^{sample} is generated from the proposal distribution $p(y^{(i)}|y^{(i-1)})$; second, based upon the proposal distribution and the full joint density π (·), the acceptance probability is computed using acceptance function $\alpha(y^{sample} \mid y^{(i-1)})$; third, the candidate sample is accepted with probability α, or rejected with probability $(1-\alpha)$. For multimedia LTE traffic considered in this work, the desired (fractal Pareto type) steady-state distributions are of the order of 0.80: 0.16: 0.04 as justified later in Section 4 with **Table 1**. So ξ=0.8, φ=0.16 and ï=0.04 are considered. An initial approximate estimate for the 3×3 Transition Probability Matrix (TPM), '*T*' is estimated using *machine learning* Metropolis-Hastings algorithm in order to provision a steady state process utilization ratio 0.80: 0.16: 0.04. The matrix '*T*' is denoted in Eq. (3).

$$T = \begin{pmatrix} 0.90 & 0.07 & 0.03 \\ 0.39 & 0.55 & 0.06 \\ 0.42 & 0.17 & 0.41 \end{pmatrix} \tag{3}$$

The Π, the state probability vector, is treated as process utilization ratio as discussed earlier. Ignoring *apriori* information, an initial unbiased state probability vector, $\Pi_0 = 1/3[1\ 1\ 1]$ is applied. The *estimated* final state probability vector, Π_f is obtained as, *[0.79818:0.16176: 0.04006]*, illustrated in **Fig. 2(a)**. **Fig. 2(b)** indicates that, although an initial biased state probability vector, $\Pi_0 = [0.1\ 0.5\ 0.4]$ is applied, the *estimated* final state probability vector, Π_f is obtained as, $\Pi_f = [0.79819:0.16148:0.04033]$, approximately same as in **Fig. 2(a)**. The result validates that a final practical process utilization ratio, $\Pi_u = [U_1 : U_2 : U_3]$ distribution i.e. [0.80: 0.16: 0.04] for three processes, has been achieved, irrespective of the initial distribution. It is to be noted that a specific value of U_i, achieved here, is under the control of designer's requirement. In general, any target v*alues* of Π_f, namely, *[0.81 0.130.06], [0.65 0.25 0.10]*, etc. can be achieved as per designer's requirement because Metropolis-Hastings algorithm can generate any arbitrary desired steady state distribution.



(a) (b)

Fig. 2. Confirmation of convergence of three states: (a) $\prod_0 = 1/3[1\ 1\ 1]$, (b) $\prod_0 = [0.1\ 0.5\ 0.4]$

4. Scheduling Mechanism and Queue Management

The work-flow of multi-service packet scheduling scheme, *QUEST for 4G LTE traffic* is visualized in **Fig. 3**. The model accepts three classes of 4G LTE traffic - Conversational voice (P_1), live streaming video (P_2), buffered streaming video & TCP based applications (P_3). The three traffic streams are classified by a traffic classifier (payload-based) and fed to three distributed FIFO ready queues: Q_1, Q_2 Q_3 for P_1, P_2, P_3, respectively. Migrations of traffic among the queues are not permissible. We define the proposed model as, M/BP/1/./QUEST. In this model, traffic arrivals are represented by 'M'. Here the arrivals are of Markovian type modulated by Poisson process (MMPP). In real world applications, this scheme is a fair estimation of large number of independent memory-less events [28]. Further, according to recent studies [29], for a settled system, incoming traffic streams defined by different distributions converge to a Poisson distribution as time evolves. The service time distribution is denoted by ‘BP’. Here, ‘BP’ denotes Bounded Pareto type. ‘1’ indicates single processor which executes processes. Maximum number of processes of multiclass traffic streams in this system including the one being serviced are denoted by ‘.’**.** The incoming processes (traffic) are being scheduled and executed according to the QUEST scheduler in a preemptive-resume manner. Processes of traffic classes are assigned and executed with priorities. The class-specific deadlines for the traffic are set stated in **Table 1.**

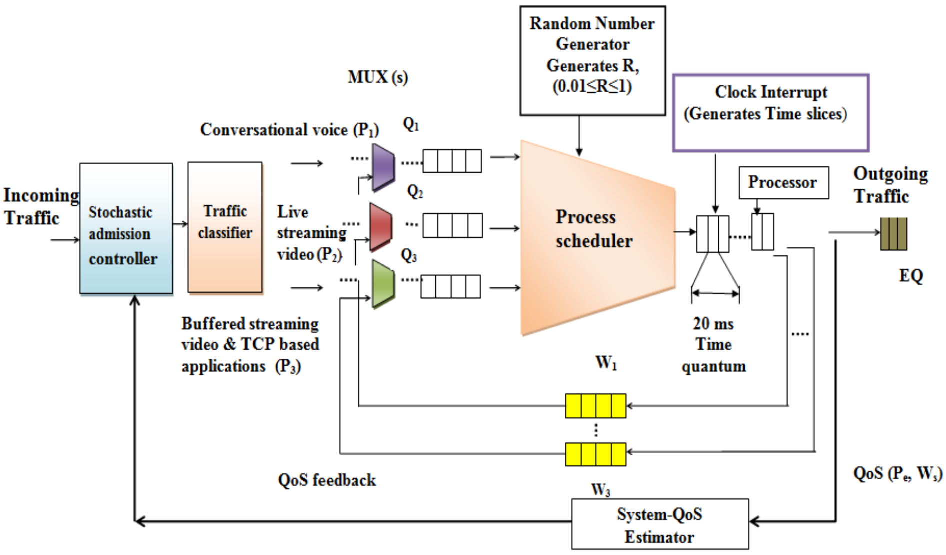

Random Number Generator Generates R, (0.01≤R≤1)
MUX (s)
Clock Interrupt (Generates Time slices)
Conversational voice (P_1)
Q_1
Q_2
Q_3
Incoming Traffic
Stochastic admission controller
Traffic classifier
Live streaming video (P_2)
Process scheduler
Processor
Outgoing Traffic
EQ
Buffered streaming video & TCP based applications (P_3)
W_1
20 ms Time quantum
QoS feedback
W_3
QoS (P_e, W_s)
System-QoS Estimator

Fig. 3. Illustration of M/BP/1/./QUEST model.
Q_i: Ready queues, W_i: Waiting queues, EQ: Expired queue.

Table 1. Service model parameters (3 classes of 4G LTE traffic)

LTE QCI	**Service class**	**Service type**	**Deadline** (ms)	**Arrival feature**
1	P_1(Conversational voice)	GBR	20	MMPP
7	P_2 (live streaming video)	Non-GBR	100	MMPP
8, 9	P_3 (buffered streaming video & TCP based applications)	Non-GBR	300	MMPP

These values of deadlines have been taken considering acceptable practical deadline of LTE traffic [30, 31] in real world applications. At times it becomes necessary to run a certain process of higher priority class before another running process of lower priority class. If a lower priority running process is interrupted for some time and resumed later after the execution of the higher priority class then the operation is called the scheduling of the processes in a preemptive-resume manner.

The priorities assigned to processes are inversely proportional to their deadline . Therefore, the priority of execution of processes are kept in the order of, $P_1 > P_2 > P_3$ and process utilization ratio is provisioned as $[U_1:U_2:U_3]$. In this scheduling policy, a clock interrupt generates the timing slices or *quanta*. After each slice, the next process is picked up from the ready queue(s). The scheduler runs through the ready queue, selects a process from a queue of processes to execute depending on the outcome of a random number generator, runs through the time slice, eventually placing the finished process in an expired queue. W_1, W_2 and W_3 denote waiting queues for the processes of traffic classes P_1, P_2, and P_3, respectively. During execution, in case any fault or interrupt occurs, the processes will be sent to respective waiting states (queues). The waiting processes multiplexed with incoming processes will be in their corresponding ready queues for further processing. A QoS feedback controller denoted as *System-QoS Estimator* with help of run-time PLR (denoted as P_e) and mean waiting time (W_s) instructs the admission controller to restrict the traffic flows. For practical real-time tasks, deadlines are in the range of 10-300 ms [9,10]. Considering uniform burst time which is made possible by traffic conditioning algorithms like token bucket, leaky bucket, etc., the process utilization (U_i) [5, 32] of the system is expressed in Eq. (4).

$$\sum_{1}^{3} U_i = T_B \cdot \left(\frac{1}{D_1} + \frac{1}{D_2} + \frac{1}{D_3} \right) \le 1 \tag{4}$$

In this scheme, T_B denotes the burst time (service time) and the deadlines of processes are denoted by D_i. In case, $D_1 = 20$ *ms*, $D_2 = 100$ *ms*, $D_3 = 400$ *ms* the value of burst time is calculated as, $T_B \le 16$ *ms*. Allowing *4 ms* timing jitter (T_J) provides the required value of time quantum (T_Q). Thus, $T_Q = T_B + T_J = 20$ *ms*. In this framework, the time quantum, T_Q , is set at 20 ms so that pre-emption does not result in deadline misses. In practical case, this value of time quantum 20 ms is acceptable because it is at least equal to the minimum process deadline 20 ms, which is required for highest priority Conversational voice (process P_1) traffic to avoid context switching. Thus, designing the value of burst time as 16 ms validates *its* use to keep the system utilization 100 percent.

4.1 Algorithm 1: QUEST Scheduling Algorithm

Algorithm 1 states formal description of proposed scheduling algorithm using Markovian property.

Algorithm1:QUEST

```
Generate random number R, $0.01 \le R \le 1$;
Set: Time quantum $T_Q$: 20 ms;
Initialize: timer, t=0
  for t=1,2...20 ms, do
    switch (initial process) {
        CASE  initial_process:$P_1$
              if ($0.01 \le R \le 0.9$) then
                  execute $P_1$;
              else if ($0.91 \le R \le 0.97$) then
                  execute $P_2$;
              else execute $P_3$;
              end if;
        CASE  initial_process:$P_2$
              if ($0.01 \le R \le 0.55$) then
                  execute $P_2$;
              else if ($0.56 \le R \le 0.94$) then
                  execute $P_1$;
              else execute $P_3$;
                  end if;
        CASE  initial_process:$P_3$
              if ($0.01 \le R \le 0.41$) then
                  execute $P_3$;
              else if ($0.42 \le R \le 0.83$) then
                  execute $P_1$;
              else execute $P_2$;
              end if;    }
  end for;
Place $P_i$ in expired queue;
```

Algorithm 1 clearly indicates that QUEST is a *true dynamic-priority* scheduler because the next process to be executed depends purely on the outcome of the random number generator decided at run-time and *may not* have the highest priority among the pending processes. Further, the next process will execute depends on which process is running currently and it is independent of the processes which were executed earlier. For example, according to the QUEST algorithm the process P_2 will run iff the outcome of random number generator is 0.94 and at present process P_1 is running.

Maximum possible value of process utilization ratio is 100 percent in the range of 1 to 100 percent. Therefore, the probability falls in the range of [0.01,1]. R is designed in such a way that it will only generate random numbers of two decimal places resolution.

4.2 Mean Waiting Time

Let, a random variable *X* taking value x in the interval *[l, q]*. The probability density function of Bounded Pareto distribution of queue service time is given by

$$f_x(x) = \frac{\theta . l^{\theta} . x^{-(\theta+1)}}{1-\left(\frac{l}{q}\right)^{\theta}}, \qquad l \le x \le q \tag{5}$$

where θ is the shape parameter, *l* and *q* denote minimum and maximum LTE data file sizes, respectively.

The second moment of this distribution is calculated as,

$$E_x(x^2) = \int_l^q x^2 . f_x(x) dx = \frac{\theta . l^{\theta}}{1-\left(\frac{l}{q}\right)^{\theta}} . \frac{\theta}{(\theta-2)} . (l^{2-\theta} - q^{2-\theta}) \tag{6}$$

The second moment of the service time distribution, *E[X^2]* is calculated as,

$$E[X^2] = \frac{E_X(x^2)}{L_C^2} \tag{7}$$

where L_c, is link capacity of the system.

From queueing theory, mean waiting time, W_s without using a stochastic admission controller, can be expressed as

$$W_S = \frac{\lambda E[X^2]}{2(1-\rho)} \tag{8}$$

where ρ is normalized load of the system.

4.3 Packet Loss Rate (PLR)

PLR generally refers to the percentage of packets that are lost during the transition from the sender to the receiver. In this work, the PLR is expressed as the root mean square error, $P_{e,rms}$ of L1 , L2 cache miss and deadline miss errors of the system. $P_{e,rms}$ is stated in Eq. (9). L1 cache miss error, L2 cache miss errors and the deadline miss errors are denoted by C_{L1}, C_{L2} and D_e respectively.

$$P_{e,rms} = \sqrt{C_{L1}^2 + C_{L2}^2 + D_e^2} \tag{9}$$

For each of the three processes: Conversational voice (P_1), live streaming video (P_2), buffered streaming video & TCP based applications (P_3), the above r.m.s error is calculated from Eq. (9) and substituted in the second row of error probability matrix, *E*, given in Eq. (10).

5. Simulation Methodology and Environment

For simulation, we consider an initial model which is characterized by two matrixes, i) the TPM, '*T*' stated in Eq. (3) for the Markov model considered (here three-state model) and ii) '*E*', an error (vector) probability matrix in (10). Practical values of cache miss errors [6, 7] and deadline miss error [8] rates have been taken.

$$E = \begin{pmatrix} 0.98 & 0.9 & 0.8 \\ 0.02 & 0.1 & 0.2 \end{pmatrix} \tag{10}$$

The three elements in the second row in Eq. (10) represent error probabilities of the processes and the elements in first row indicate the probabilities of correctness. The simulation framework has been developed in the enviornment of a discrete event and discrete time simulation tool, DEVS suite [33] and MATLAB R 2015 b (8.6) in a laptop having specification of Intel i3 CPU 2.5 GHz, 4 GB RAM, Windows 7 platform. DEVS suite supports animation with I/O and state trajectories of computer network models. The simulator offers high-level model abstraction. Monte Carlo simulation method has been applied (using Statistics and Machine Learning Toolbox™ of MATLAB) for confirmation. Following (**Table 2**) system environment for simulation was used:

Table 2. System parameters used for simulation

Parameter	Conditions
Arrival rate	50 packets/s
Data file size	20-400 KB
Burst time	16 ms
Shape parameter (θ)	0.14
Service discipline	QUEST
Link Capacity (L_c)	10 Mbps
Packet size	1 KB
Simulation time	439 s

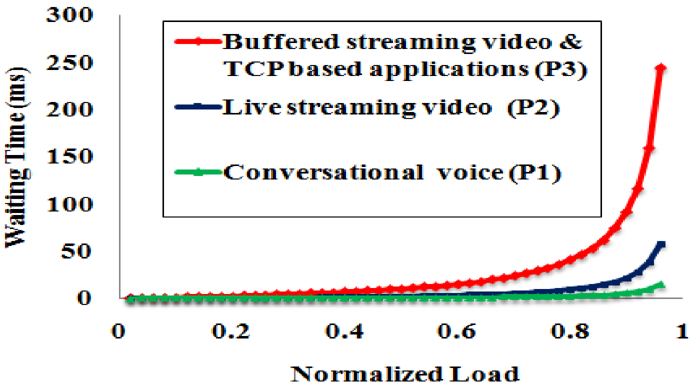


Fig. 4. Waiting time comparison for different processes for QUEST

6.Simulation Results

The performance of the scheduler in terms of waiting time of individual flows, mean waitinng time over normalized load and convergence of state probability vector have been discussed in this section.

6.1 Waiting Time of Individual Process

Waiting time for each class of traffic in this simulation are plotted with respect to increasing normalized load shown in **Fig. 4**.

6.2 Performance Analysis of Mean Waiting Time

In this subsection we compare the performance of the QUEST scheduler for the QoS parameter mean waiting time with current benchmark scheduling algorithms - deferred pre-emption (DP) [34] , earliest deadline first (EDF) [32] and accuracy-aware EDF (A-EDF) [35]. The results have been shown in **Fig. 5**. The individual process loads are distributed in the ratio of 0.80: 0.16:0.04 for a normalized load. For example, if the normalized load is 0.5, then the individual process load would be in the ratio of 0.4:0.08:0.02.

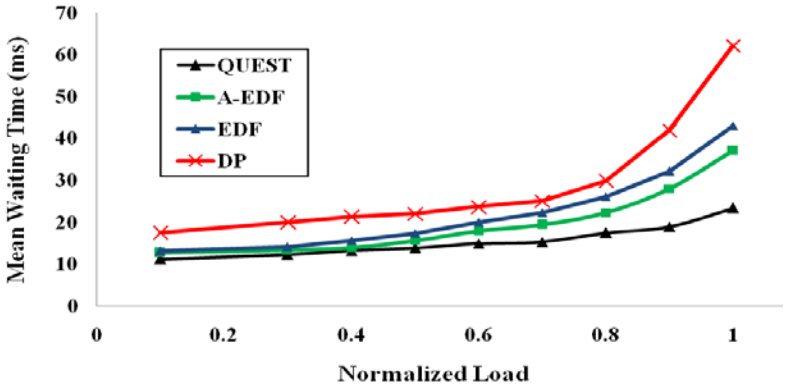


Fig. 5. Mean waiting time with increasing load.

Fig. 5 illustrates that the QUEST scheduler has lowest value of mean waiting time with increasing normalized load and it shows 23 percent performance improvement with respect to A-EDF scheduler. In QUEST stochastic admission controller [36] is used. This is permissible in QUEST and keeps the mean waiting time low even at rise of traffic loads close to 100 percent. On the other hand, EDF and its variant A-EDF are not stochastic, avoiding usage of such admission controllers. Therefore, for EDF, mean waiting time can be low only for loads below about 80 percent [36], which contradicts our original problem objective of 100 percent utilization. If stochastic admission controller is not used, in high load condition, the rise-rate of mean waiting time would be steeper as happens with EDF and A-EDF scheduler shown in **Fig. 5**. Moreover, RM (rate monotonic) and DP schedulers (**Fig. 5**) are static priority based and therefore, have significant rise of mean waiting time with increase of load.

6.3 Estimation of Steady State Probability Analysis and System Stability

Simulation was performed considering random arrival of processes (traffic) with the given error vector. The error vector provides error positions in 900 (iterations). The probability of finding the processor in a given state is calculated from ‘*T*’. Further we obtain the error probability from the matrix ‘*E*’.

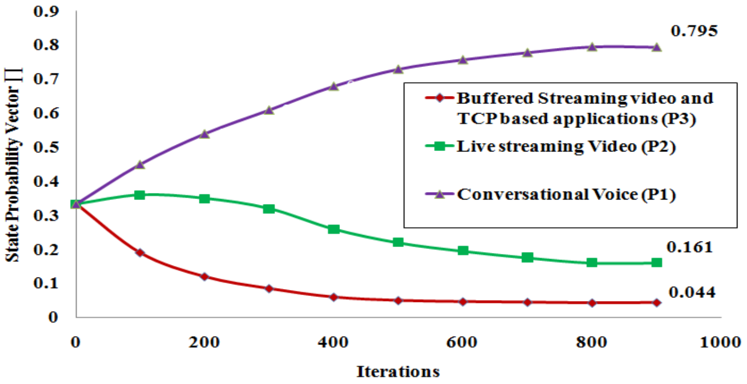


Fig. 6. Convergence of State Probability Vector Π.

As shown in **Fig. 6**, Process P_1 (Conversational voice), Process P_2 (Live streaming video), and Process P_3 (Buffered streaming video and TCP based applications) achieve a steady state probability vector of process utilization ratio of 0.795, 0.161 and 0.044, respectively and the PLR (denoted as P_e) comes as 0.00451 (see **Fig. 7**). This value of PLR is acceptable because it falls within the standard PLR threshold of 1 percent [37].

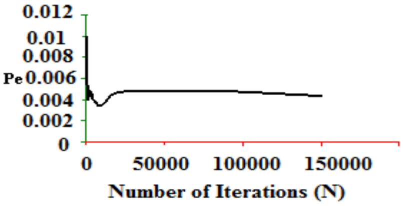


Fig. 7. P_e converges to a steady state with number of increasing iterations

Thus, the lowest priority process traffic buffered streaming video and TCP based applications secures a *guaranteed* 4.4 percent process utilization. This validates the authors' claim that *low-priority process (traffic) starvation* is *eliminated*. Simulation is performed to calculate the packet loss rate which is denoted as P_e. **Fig. 7** shows that with the increasing count of sequences (iterations), P_e settles to a steady state value. This validates consideration of the processes as stable Markov states, and establishes system stability.

7. Dynamic global optimization and re-configurability

Our objective is to *minimize the PLR* in order to maximize system QoS. As the run-time load varies in an LTE router, the pre-allocated state transition probabilities of matrix '*T*' (stated in Eq. 3) are unfit to provision the QoS at its *maximum* value. This problem is *solved* by re-configuring the initial Transition Probability Matrix (TPM), '*T*' using reconfiguration (*tuning*) parameters, Δ_1, Δ_2 and Δ_3 stated in Eq. (11).

$$T_{recon} = \begin{pmatrix} 0.90-2\Delta_1 & 0.07+\Delta_1 & 0.03+\Delta_1 \\ 0.39+\Delta_2 & 0.55-2\Delta_2 & 0.06+\Delta_2 \\ 0.42+\Delta_3 & 0.17+\Delta_3 & 0.41-2\Delta_3 \end{pmatrix} \tag{11}$$

These reconfiguration parameters drive the PLR to a *minimum* value and hence QoS back to maximum value by the feedback controller shown in **Fig. 8**.

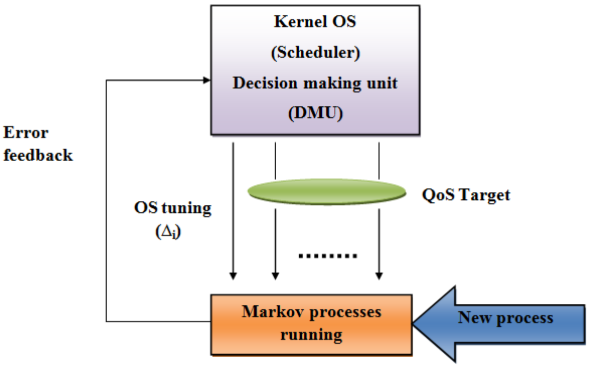


Fig. 8. Feedback controller for re-configuring QUEST

In practical cases, the processor usage allotment to all processes is dynamic over time and based on events. The system QoS is dynamically monitored by the scheduler using a feedback controller with the help of decision making unit (DMU) and necessary corrections are executed.

The advantages of using feedback controller in the proposed scheduler are as follows. Feedback controller increases performance of QUEST irrespective of internal and external uncertainties. Moreover, it reconfigures the proposed QUEST to run within user-defined range on the fly.

The error feedback controller is used to reconfigure the QUEST by suitably tuning Δ_i s. The 3D-contour plot of PLR (denoted as P_e) as function of Δ_1 and Δ_2 with $\Delta_3 = 0$) is shown in **Fig. 9**. Similarly, P_e can be plotted as function of Δ_2 , Δ_3 and Δ_1 , Δ_3. It has been noted that P_e is *globally minimum* at 0.0012 if values of Δ_1, Δ_2, Δ_3 are kept at 0.0251, -0.089 and 0, respectively.

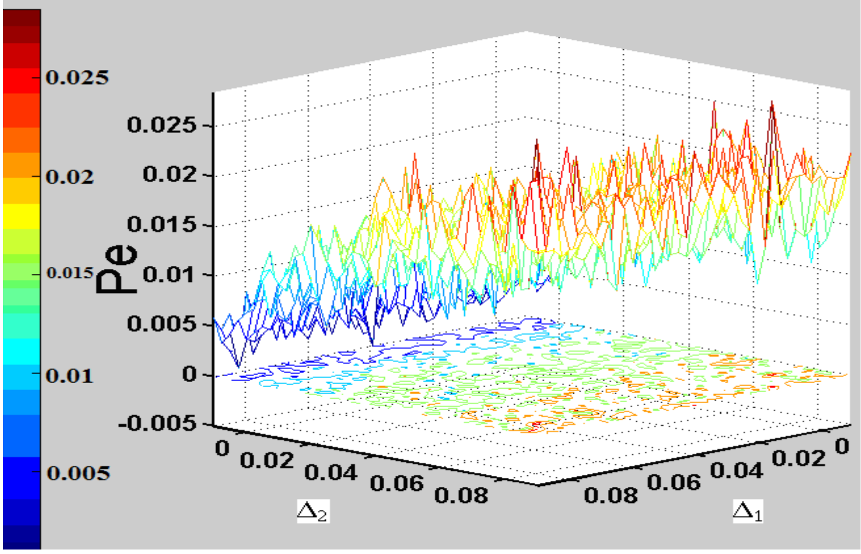


Fig. 9. 3D plot of P_e with respect to Δ_1, Δ_2; $\Delta_3 = 0$

8. Run-time estimation of TPM by machine learning

Because of the *re-configurable* property of the QUEST scheduler, specific values of TPM parameters at a given time during system operation are *uncertain*. Therefore, it is important to *dynamically estimate* the TPM parameters (elements of the matrix '*T*') during run-time. The transition probability matrix (TPM) parameters are estimated by a forward-backward (Baum-Welch) *machine-learning algorithm, which* learns during *run-time* from the observed error patterns (sequences). The error patterns serve as *training* data. Here, for a given Δ_i algorithm 3 is applied to estimate the TPM parameters. The algorithm is illustrated as follows.

Algorithm 3 Forward-backward (baum welch) machine learning

```
1.  begin set: iteration index $n \leftarrow 0$
      initialize:
            Transition probability $p_{ij}$,
            Probability of error $e_{jk}$,
            Training sequence $S^{T}$
            define: convergence criterion $\dot{c}$
2.        do $n \leftarrow n+1$
3.          Compute:
              probability of transition $p'(n)$ from $p(n-1)$ and $e(n-1)$                      using (14)
              improved estimate of probability of error $e'(n)$ from $p(n-1)$ and $e(n-1)$   using (15)
4.          update values:
              $p_{ij}(n) \leftarrow p'_{ij}(n-1)$
              $e_{jk}(n) \leftarrow e'_{jk}(n-1)$
5.        until $w < \dot{c}$   (Convergence achieved)
              where  $w = \max_{i,j,k} \{p_{ij}(n) - p_{ij}(n-1), e_{jk}(n) - e_{jk}(n-1)\}$
6.    return:
            $p_{ij} \leftarrow p_{ij}(n)$
            $e_{jk} \leftarrow e'_{jk}(n)$
7.  end
```

In this algorithm, p_{ij}, e_{jk} and n are given by transition probability, probability of error and iteration index, respectively. Let, $\ddot{w}_i(t)$ and $\ddot{w}_i(t+1)$, denote the current state and the next state of the FSM respectively. The *visible* error pattern is presented by S=[$010^{2}0..1000^{3}01..0^{4}00$] where elements of this pattern are denoted by S_k and 1s represent errors. In this pattern the superscript form 0^{2},, 0^{3} and 0^{4} denote consecutive two ("00"), three ("000") and four ("0000") zeros, respectively. The dots (..) present continuous stream of patterns (continuous 0 and 1 s). In the pattern the zero (es) in preceding and following superscript have been stated in order to present the continuity of the pattern. Therefore, the pattern S=[01000..10000001..000000] is represented in the compact form as, S=[$010^{2}0..1000^{3}01..0^{4}00$]. In a more compact form the pattern may be presnted as, S=[$010^{3}..10^{5}01..0^{5}0$]. Alternatively, the pattern can be represented in a most compact form as, S=[$010^{n}..10^{m}1..0^{k}$]. In this case, n=3, m=6, and k=6. Further, it may be noted that, representation of a sequence of 0 s and 1 s in a compact form is only subject to convenience and there is no fixed hard and fast rule. Representation in any given form does not change any result of the paper and therefore is not much critically important. All representations give the same result.

We have, $$p_{ij}=P[\ddot{w}_j(t+1)|\ \ddot{w}_i(t)] \qquad (12)$$

and $$e_{jk}=P[S_k(t)|\ddot{w}_j(t)] \qquad (13)$$

Computation has been started with an estimate of p_{ij} and e_{jk} and to calculate improved values of them until convergence criterion, $\dot{c}$ is achieved. In this estimation, $x_i(t)$ is the probability that the scheduler is in state $\ddot{w}_i(t)$ and has generated the error sequence up to step t. Similarly, $y_i(t)$

to be the probability that the model is in state $\ddot{w}_i(t)$ and will generate the rest of the error sequence. An improved value can be calculated by defining $z_{ij}(t)$ - the probability of transition between $\ddot{w}_i\,(t-1)$ and $\ddot{w}_j(t)$, given the model generated the entire training visible sequence S^T by any path. The $z_{ij}(t)$ is represented in Eq. (14).

$$z_{ij}(t) = \frac{p_{ij} e_{jk} x_i(t-1) y_j(t)}{P(S^T \mid \dot{c})} \tag{14}$$

In Eq. (14), the $P(S^T\,|\dot{c})$ denotes the probability that the model generated sequence S^T. Let, p'_{ij} is the estimate of the probability of a transition from $\ddot{w}_i(t-1)$ to $\ddot{w}_j(t)$. The value of p'_{ij} can be calculated by taking the ratio between the expected number of transitions from $\ddot{w}_i$ to $\ddot{w}_j$ and the total expected number of transitions from $\ddot{w}_i$.

$$p'_{ij}(t) = \frac{\sum_{t=1}^{T} z_{ij}(t)}{\sum_{1}^{T} \sum_{k} z_{ik}(t)} \tag{15}$$

An improved estimation of e'_{jk} can be calculated,

$$e'_{jk}(t) = \frac{\sum_{\substack{t=1 \\ s(t)=s_k}}^{T} \sum_{l} z_{jl}(t)}{\sum_{t=1}^{T} \sum_{l} z_{jl}(t)} \tag{16}$$

Improved estimates for p_{ij} and e_{jk} are repeated using Eq. (15) and Eq. (16) until the change is significantly less than convergence criterion ċ. In this estimation, ċ has been set at 0.001.

8.1 Stability and Accuracy of Run-time TPM Estimation

In real world application, the process load varies dynamically within a network router resulting variation of PLR. Therefore, the elements of 'E', the error probability matrix also changes with respect to time and iterations. The system simulates the newly estimated model having modified TPM. **Fig. 10** shows that in this *learning*, forward-backward algorithm is guaranteed to converge to a maximum log likelihood ratio.

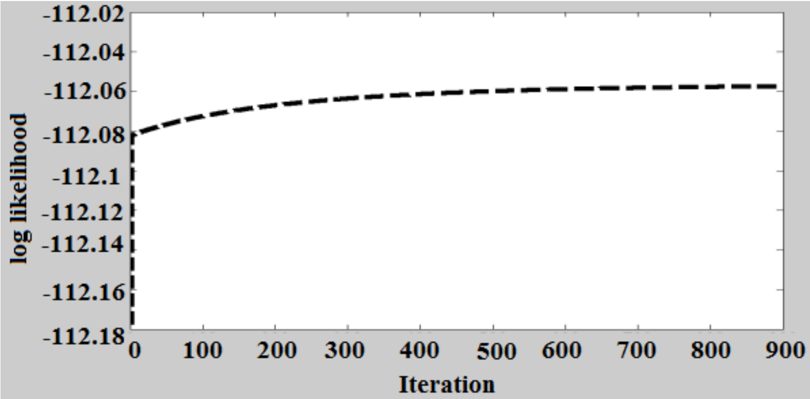


Fig. 10. Log likelihood with respect to increasing turns (iterations).

This convergence signifies stability of the system.

In order to validate the accuracy of the proposed scheduler we compare the run-time error patterns for initially considered TPM and for the estimated regenerated one. These patterns are depicted in **Fig. 11**.

From the figure it is clear that, the two run-time error patterns are almost *identical*. This validates accuracy of the proposed model.

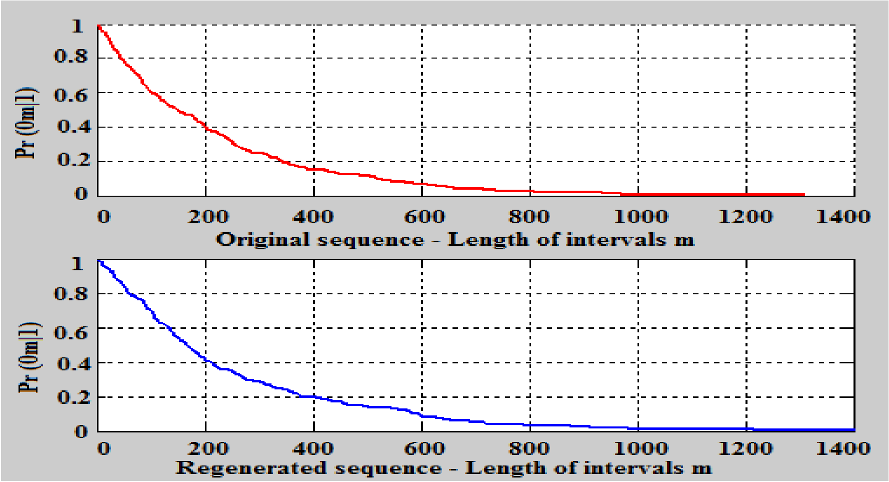


Fig. 11. Pr (0^m|1) for initial and regenerated model.

9. Performance analysis of QUEST

During execution, PLRs for individual traffic flow in QUEST are depicted in **Fig. 12**.

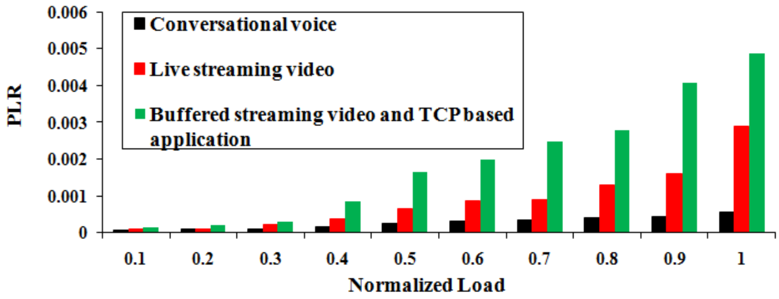


Fig. 12. PLR for each LTE traffic

The figure indicates that the conversational voice traffic flow in QUEST has a minimum value of PLR with increasing normalized load compared to other flows. The rise rate of run-time PLR for lowest priority buffered streaming video and TCP based application traffic is significantly highest.

The performance of the proposed scheduler is compared with the performance of current benchmark scheduling algorithms - deferred preemption (DP), earliest deadline first (EDF), accuracy-aware EDF (A-EDF) for increasing normalized loads. The results are plotted in **Fig. 13**. In this simulation the individual process loads are distributed in the ratio of 0.80: 0.16:0.04 for a normalized load. For example, if the normalized load is 0.5, then the individual process load would be in the ratio of 0.4:0.08:0.02.

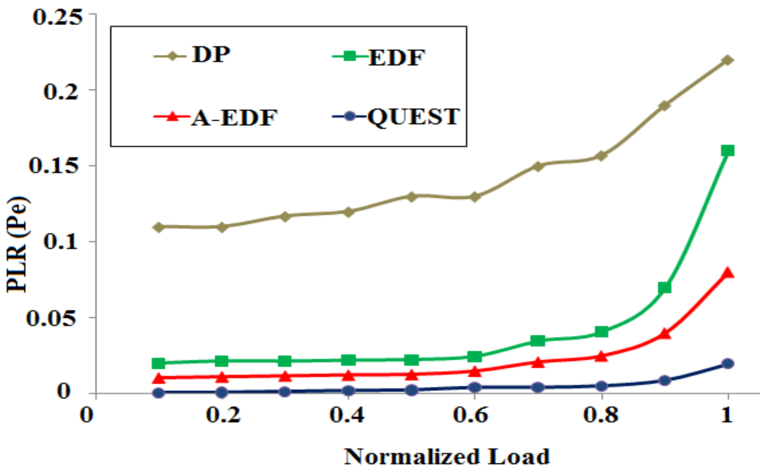


Fig. 13. PLR for DP, EDF, A-EDF, QUEST.

Fig. 14 illustrates cache miss errors (L1, L2) and deadline miss errors for aforementioned scheduling algorithms with typical values of L1=32 KBytes and L2=256 KBytes at a normalized load of 0.9.

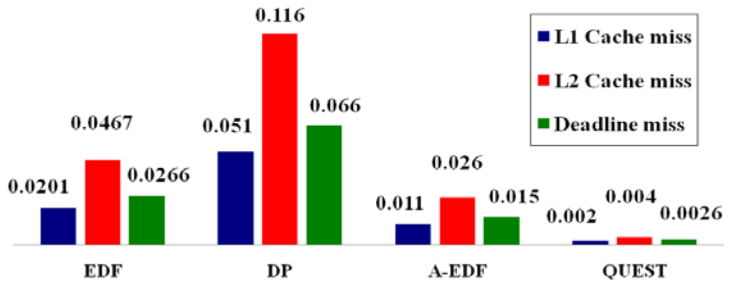


Fig. 14. Cache and deadline miss errors for DP, EDF, A-EDF and QUEST.

Figs. 13 and **14** indicate that QUEST scheduler outperforms other scheduling schemes and it has a lowest value of PLR. In QUEST, the packet loss rate is reduced by 37 percent compared to A-EDF with lower values of cache and deadline miss errors. The performance improvement achieved in QUEST is due to use of Hidden Markov Model (HMM) filter (Baum-Welch based) which is a probabilistic model applicable for finite and discrete process states. On the other hand, A-EDF uses Kalman filter for process state estimation. Kalman filter is a special case of HMM applicable only for continuous and infinite states for a linear state space model, which is not valid in digital embedded systems. Moreover, Kalman filter assumes Gaussian noise, whereas HMM filter makes no such assumptions and is thus more *general and accurate*. Further, EDF and A-EDF have no explicit control on utilization. As a result, both the schedulers experience *high* deadline miss rates at *heavy loads*. In contrast, during run-time QUEST enforces utilization close to 100 percent, with lower deadline miss rates even at heavy traffic loads. This establishes QUEST's superiority of performance over A-EDF and EDF.

10. Experimental Setup

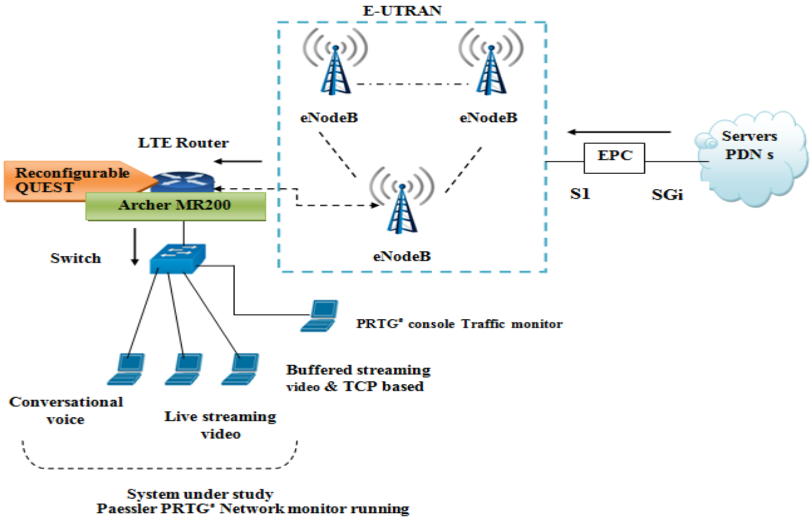
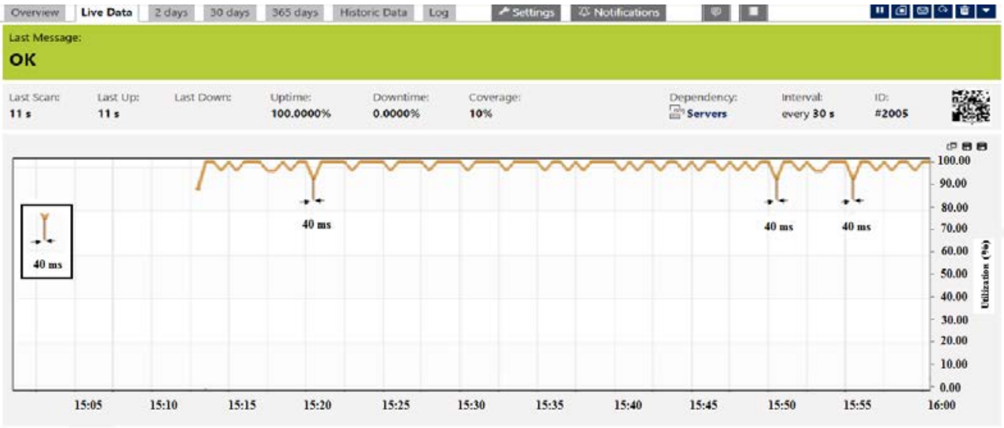


Fig. 15. Experimental setup for QUEST scheduler

PDN: Packet Data Network; SGi, S1: Interfaces; EPC: Evolved Packet Core; eNodeB: Evolved Base Stations; E-UTRAN: Evolved UMTS Terrestrial Radio Access Network

The performance of the proposed QUEST scheduler was validated in a TP-Link Archer™ MR 200 platform [38] of 4G LTE network router. The platform was customized for the implementation of the reconfigurable scheduler QUEST in the following experimental setup shown in **Fig. 15**.

Three classes of multimedia 4G LTE traffic, namely, conversational voice, live streaming video and buffered streaming video and TCP based were being scheduled and executed according to the QUEST. Three laptops were used for receiving each class of traffic and a renowned Paessler PRTG™ network monitor [39] console was connected with a multiport switch to monitor the performance of the QUEST. The *trace* of the run-time processor utilization and the trtansient response (time trace) of the feedback controller over a continuous monitor of 93 minutes is presented in **Fig. 16**.

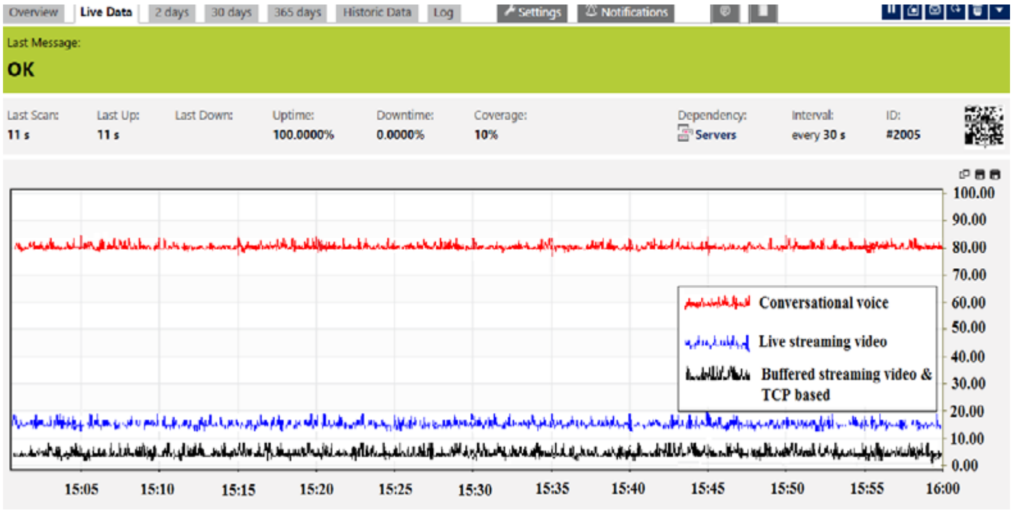


Last Message:
OK

Last Scan: 11 s
Last Up: 11 s
Last Down:
Uptime: 100.0000%
Downtime: 0.0000%
Coverage: 10%
Dependency: Servers
Interval: every 30 s
ID: #2005

Fig. 16. Transient response (time trace) of the feedback controller in QUEST scheduler

As shown in the figure, a utilization close to 100 percent (within the range of 92 to 97 percent) was maintained. However, when a heavy data burst arrives, the utilizatiom falls below 92 percent and the transient response of feedback controller of the QUEST scheduler has a settling time ≈40 ms, shown in the figure. This value of settling time is within the industrial specification limit of 50 ms [40]. This is because of QUEST's unique advantage that the utilization is fixed nearly at 100 percent.



Last Message:
OK

Last Scan: 11 s
Last Up: 11 s
Last Down:
Uptime: 100.0000%
Downtime: 0.0000%
Coverage: 10%
Dependency: Servers
Interval: every 30 s
ID: #2005

Fig. 17. Time trace of process utilization ratio for conversational voice, live streaming video and buffered streaming video and TCP based traffic

The experimental results (depicted in **Fig. 17**) indicate that the steady state process utilization ratio in the order of 80:16:4 for conversational voice, live streaming video, buffered streaming video and TCP based traffic has been achieved. Thus, the lowest priority process traffic flow, buffered streaming video and TCP based applications secures a *guaranteed* 4 percent process utilization *avoiding low-priority process* (traffic class) *starvation.*

11. Conclusion

In this paper we present and investigate on a novel re-configurable QoS-enhanced intelligent real-time packet scheduler - QUEST, for multimedia 4G LTE traffic in routers. Machine learning algorithms were used for the first time to our best knowledge to design a QoS-maximized optimal fair stochastic packet scheduler to dynamically optimize the system QoS during run-time. In contrast to the existing schedulers proposed in the literature, this scheduler was demonstrated to maximize the system-QoS, guaranteeing utilization fixed nearly at 100 percent. Further, the proposed scheduler has following unique advantages. It addresses poor performance of the premier benchmark schedulers : DP, EDF and A-EDF at heavy loads. Moreover, QUEST avoids the problem of priority starvation for low priority processes: buffered streaming video and TCP based applications and offers the benefit of arbitrary pre-programming of process utilization ratio.

Two important QoS metrics, PLR and mean waiting time (related to system latency) were studied. Simulation results show that the proposed scheduler, QUEST in a LTE platform performs significantly better compared with existing benchmark schedulers. Further, the study indicate that scheduler has achieved an improvement of 37 percent in PLR and an improvement of 23 percent in mean waiting time over the current state-of-the-art benchmark scheduler A-EDF. The accuracy of the QUEST was validated by comparing the run-time error patterns for initial and estimated TPM. The proposed QUEST was implemented and validated in a customized TP-Link Archer™ MR 200 4G LTE router platform and results were analyzed using Paessler PRTG™ network monitor. The experimental results indicate that a guaranteed steady state process utilization ratio in the order of 80:16:4 for conversational voice, live streaming video, buffered streaming video and TCP based traffic has been achieved and maintained. Further we observe that, during run-time the transient response of the feedback controller in QUEST has a short settling time ≈40 ms, which is within the real-time industrial specification limit of 50 ms.

References

[1] X. Zhou, J. Wei and C.-Z Xu, "Quality-of-service differentiation on the Internet: A taxonomy," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 354–383, January, 2007. Article (CrossRef Link)

[2] Q. Liu, R. Hu and S Liu, "A wireless location system in LTE networks," *Mobile Information Systems*, vol. 2017, article ID 6160489, pp. 1-11, February, 2017. Article (CrossRef Link)

[3] N. Larasati, W. K. Kwee, S. C. Chong and Y. Wee, "An analysis on quality of service enhancement in Long Term Evolution networks: past, present and future," *Middle-East Journal of Scientific Research* , vol. 24, no. 3, pp. 498-513, 2016. Article (CrossRef Link)

[4] P. Ameigeiras, J. Navarro-Ortiz1, P. Andres-Maldonado, J. M. Lopez-Soler, J. Lorca, Q. Perez-Tarrero and R. Garcia-Perez, "3GPP QoS-based scheduling framework for LTE," *EURASIP Journal on Wireless Communications*, vol. 2016, pp. 1-14, March, 2016. Article (CrossRef Link)

[5] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of ACM*, vol. 20, no. 1, pp. 46-61, January, 1973. Article (CrossRef Link)

[6] D. Thiébaut, J. L. Wolf and H. S. Stone, "Synthetic traces for trace-driven simulation of cache memories," *IEEE Transactions on Computers*, vol. 41, no. 4, pp. 388-410, April, 1992. Article (CrossRef Link)

[7] J. P. Singh, H. S. Stone and D. F. Thiebaut, "A model of workloads and its use in miss-rate prediction for fully associative caches," *IEEE Transactions on Computer*, vol. 41, no. 7, pp. 811-825, July, 1992. Article (CrossRef Link)

[8] K.-D. Kang, S. H. Son, J. A. Stankovic, "Managing deadline miss ratio and sensor data freshness in real-time databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1200-1216, October, 2004. Article (CrossRef Link)

[9] M. R. Tabany, C. G. Guy and R. S. Sherratt, "A novel downlink semi-persistent packet scheduling scheme for VoLTE traffic over heterogeneous wireless networks," *EURASIP Journal on Wireless Communications*, vol. 2017, pp. 1-14, April, 2017. Article (CrossRef Link)

[10] S. Chen and K. Nahrstedt, "An overview of quality of service routing for the next generation high-speed networks: problems and solutions," *IEEE Network*, vol. 12, no. 6, pp. 64-79, November/December 1998. Article (CrossRef Link)

[11] S. Wang, D. Xuan, R. Bettati and W. Zhao, "Toward statistical QoS guarantees in a differentiated services network," *Springer Telecommunication Systems*, vol. 43, no. 3-4, pp. 253–263, April, 2010. Article (CrossRef Link)

[12] L. Gavrilovska and D. Talevski, "Novel scheduling algorithms for LTE downlink transmission," in *Proc. of 9th Telecommunications Forum (TELFOR)*, pp. 398–401, November 22-24, 2011. Article (CrossRef Link)

[13] M. Iturralde, T. A. Yahiya, A. Wei and A.-L. Beylot, "Resource allocation using Shapley value in LTE networks," in *Proc. of 22nd International Symposium on Personal Indoor and Mobile Radio* Communications, pp. 31–35, September 11-14, 2011. Article (CrossRef Link)

[14] G. Piro, L. A. Grieco, G. Boggia, R. Fortuna and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 1052-1065, October, 2011. Article (CrossRef Link)

[15] S.-J. Wu and L. Chu, "A novel packet scheduling scheme for downlink LTE system," in *Proc. of 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 25-28, October 14-16, 2011. Article (CrossRef Link)

[16] B. Sadiq, R. Madan and A. Sampath, "Downlink scheduling for multiclass traffic in LTE," *EURASIP Journal on Wireless Communications and Networking*, Article ID 510617, pp. 1-18, December, 2009. Article (CrossRef Link)

[17] A. Marinčić and D. Šimunić, "Performance evaluation of different scheduling algorithms in LTE systems," in *Proc. of 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 595-600, May 30-June 3, 2016. Article (CrossRef Link)

[18] D.-H. Nguyen, H. Nguyen and É. Renault, "WEMQS: A new LTE downlink scheduling scheme for voice services based on user perception," *International Journal of Computer Applications*, vol. 142, no. 10, pp. 28-36, May, 2016. Article (CrossRef Link)

[19] H. Toral-Cruz, A.-S. K. Pathan, J. C. R. Pacheco, "Accurate modeling of VoIP traffic QoS parameters in current and future networks with multifractal and Markov models," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, pp. 2832-2845, June, 2013. Article (CrossRef Link)

[20] N. D. Cristofaro, G. McGill, A. Sallahi, M. Davis, A. Alsibai and M. St-Hilaire, "QoS evaluation of a voice over IP network with video: A case study," in *Proc. of Canadian Conference on Electrical and Computer Engineering*, St. John's, NL, Canada, pp. 288–292, May 3-6, 2009. Article (CrossRef Link)

[21] C. Ghazel and L. Saïdane, "Satisfying QoS requirements in NGN networks using a dynamic adaptive queuing delay control method," in *Proc. of 10th International Conference on Future Networks and Communications*, *Procedia Computer Science*, vol. 56, pp. 225-232, August 17-20, 2015. Article (CrossRef Link)

[22] L. Greco, D. Fontanelli and A. Bicchi, "Design and stability analysis for anytime control via stochastic scheduling," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 571-585, March, 2011. Article (CrossRef Link)

[23] N.-E. Rikli and S. Almogari, "Efficient priority schemes for the provision of end-to-end quality of service for multimedia traffic over MPLS VPN networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 25, no. 1, pp.89-98, January, 2013. Article (CrossRef Link)

[24] H. Kooti, D. Mishra and E. Bozorgzadeh, "Reconfiguration-aware real-time scheduling under QoS constraint," in *Proc. of 16th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 141-146, January 25-28, 2011. Article (CrossRef Link)

[25] R. B. Lyngsø and C. N. S. Pedersen, " Complexity of comparing Hidden Markov models," in *Proc. of 12th Int. Symp. on Algorithms and Computation (ISAAC), New Zealand, Springer Berlin Heidelberg*, pp. 416-428, 2001. Article (CrossRef Link)

[26] S. Chib and E. Greenberg, "General understanding the Metropolis-Hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327-335, February, 2012. Article (CrossRef Link)

[27] G. Wang, "ML estimation of transition probabilities in Jump Markov systems via Convex optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp.1492-1502, July, 2010. Article (CrossRef Link)

[28] L. Kleinrock, *Queueing Systems Theory*, Wiley, Hoboken, New Jersey, USA, vol. 1, January, 1975. Article (CrossRef Link)

[29] V. G. Abhaya, Z. Tari, P. Zeephongsekul and A. Y. Zomaya, "Performance analysis of EDF scheduling in a multi-priority preemptive M/G/1 queue," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2149-2158, August, 2014. Article (CrossRef Link)

[30] TS 23.203 V8.11.0. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects: Policy and Charging Control architecture (Rel. 8),2010. Article (CrossRef Link)

[31] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of service in LANs, WANs, and VPNs*, Cisco Press, USA, November, 2004. Article (CrossRef Link)

[32] X. Wang, I. Khemaissia, M. Khalgui, Z. W. Li, Z. Li, O. Mosbahi and M. Zhou, "Dynamic low-power Reconfiguration of real-time systems with periodic and probabilistic tasks," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 258-271, January, 2015. Article (CrossRef Link)

[33] DEVS suite Discrete event system simulator suite, Arizona Center of Integrative Modeling and Simulation of Arizona State University. https://acims.asu.edu/software/devs-suite/

[34] R. J. Bril, J. J. Lukkien and W. F. J. Verhaegh, "Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption revisited," in *Proc. of 19th Euromicro Conf. Real-Time System*, pp. 269-279, July 4-6, 2007. Article (CrossRef Link)

[35] M. Nasri, M. Kargahi and M. Mohaqeqi, "Scheduling of accuracy-constrained real-time systems in dynamic environments," *IEEE Embedded Systems Letters*, vol. 4, no. 3, pp. 61-64, September, 2012. Article (CrossRef Link)

[36] M. Ghaderi, R. Boutaba and G. W. Kenward, "Stochastic admission control for quality of service in wireless packet networks," *Lecture Notes in Computer Science Series*, vol. 3462, pp. 1309-1320, May, 2005. Article (CrossRef Link)

[37] J. Johnston, S. Farrington, R. Saville and T. Szigeti, *Medianet Reference Guide*, Cisco, pp.12-13, October, 2010. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/Medianet_Ref_Gd/medianet_ref_gd.pdf

[38] Whitepaper of TP-Link Archer™ MR 200 platform. Article (CrossRef Link)

[39] Paessler PRTG® network monitor. Article (CrossRef Link)

[40] M. Liotine, *Mission-Critical Network Planning*, Artech House, Boston – London, 2003. Article (CrossRef Link)

Suman Paul is currently an Assistant Professor, Department of Electronics and Communication Engineering, Haldia Institute of Technology, West Bengal University of Technology (Maulana Abul Kalam Azad University of Technology West Bengal), India. He received his bachelor’s and master’s degree in electronics and communication engineering and computer science, respectively in 2005 and 2008, respectively both from WBUT. He interests in scheduling and QoS in communication networks. He worked as associate researcher in the Indian Institute of Management (IIM), Calcutta in 2007 and qualified Cisco Certified Network Associate.

Malay Kumar Pandit is currently a Professor in the Electronics and Communication Engg Dept. of the Haldia Institute of Technology, Haldia, India. He received his B.E and M. E degrees in Electronics Engineering from Electronics and Telecom Engg. Dept, Jadavpur University, India in 1989 and 1991, respectively. Dr. Pandit received PhD from Cambridge University in 1996 and post-doc from the Optoelectronics Research Centre, City University of Hong Kong till 2002 where he pioneered the use of polymers for optical waveguide applications. He has eight years of industry experience. He interests in the field of scheduling and QoS issues in telecommunication networks and embedded systems.